Design of a Secure Mutually Authenticated Key-Agreement Protocol for Multi-server Architecture

Alavalapati Goutham Reddy[1], Eun-Jun Yoon[2], Young-Ju Kim[1], Kee-Young Yoo[1*]

[1] School of Computer Science and Engineering, Kyungpook National University, Daegu, Republic of Korea.
[2] Department of Cyber Security, Kyungil University, Gyeongbuk, Republic of Korea.

* Corresponding author. Email: yook@knu.ac.kr
Manuscript submitted January 5, 2017; accepted April 20, 2017.
doi: 10.17706/jcp.13.2.194-203

Abstract: Authentication with key-agreement protocols for multi-server architecture are emerging as a solution to conquer the traditional client-server architecture's limitations such as repeated registrations with distinct tokens and credentials. Since Li *et al.*'s first proposed authentication protocol for multi-server architecture, several liken protocols have tailed this queue. Majority of these protocols have been designed while the users sharing their plain or digested credentials with the servers during either registration or authentication phases. This weakens the security by making it vulnerable to severe security threats called privileged insider attacks, user impersonation attacks and server impersonation attacks. To overcome the aforementioned problems, this paper put forwards an authentication with key-agreement protocol for multi-server architecture based on biometrics. The proposed protocol is absolutely light-weight due to its design mainly based on one-way hash function. The analysis section of this paper shows that the proposed protocol performs better than related protocols and makes it suitable for practical applications.

Key words: Authentication, key-agreement, protocol, multi-server, three-factor, security, performance.

1. Introduction

The vast expansion of internet and ubiquitous computing technologies have necessitated the authentication of every remote user. Cryptographic authentication is a secure practice of transferring credentials to determine someone, in fact, who they are proclaimed to be and providing authorization to access the services subsequently. Typical authentication can be obtained in distinctive ways namely knowledge factors (passwords), possession factors (tokens) and inherence factors (biometrics) are some well-known methods. Several authors designed authentication protocols for multi-server environment using either two of the above factors or all the three factors [1]-[22]. This paper discusses the recently proposed three-factor authentication protocols under the hypothesis of biometrics are more robust than passwords and smartcards.

Related works: In 2010, Yang *et al.* [1] introduced a biometric password-based multi-server authentication protocol with smartcards. Their protocol requires lots of computations and is prone to insider attacks. In 2011, Yoon *et al.* [2] put forward a three-factor authentication protocol for multi-server architecture based on elliptic curve cryptography. Later on, He in 2011 [3] & Kim *et al.* [4] in 2012 proved that Yoon *et al.*'s protocol cannot resist masquerade attacks, insider attacks, stolen smartcard attacks and off-line password guessing attacks. Kim *et al.* [4] further proposed a biometric based authentication protocol for multi-server protocol, which was found to be lacking user anonymity and correctness in the login and password changing

phases. In 2014, Chuang *et al.* [5] proposed an anonymous three-factor multi-server authenticated key agreement protocol. Their protocol is constructed mainly on one-way hash function which makes it suitable for real time applications. Unfortunately, Mishra *et al.* [6] in 2014 & Lin *et al.* [7] in 2015 pointed out several weaknesses of Chuang *et al.*'s protocol such as lack of user anonymity, susceptible to server spoofing attacks, stolen smartcard attacks, user impersonation attacks, denial-of-service attacks and session-key compromise. Then they proposed an improved protocols over Chuang et al.'s protocol. However, Lu *et al.* [8] in 2015 & Wang *et al.* [9] in 2016 stated that Mishra *et al.*'s protocol is exposed to user and server masquerading attacks, replay attacks, forgery attacks, denial-of-service attacks, lack of perfect forward secrecy and user anonymity. Lu *et al.* [8] in 2015 & Wang *et al.* [9] in 2016 proposed improved protocols over Mishra *et al.*'s protocol. In 2015, Jiang *et al.* [10], He *et al.* [11] & Odelu *et al.* [12] put forward various authenticated key-agreement protocols for multi-server environment based on biometrics. Above three protocol involves registration center in the authentication phase which makes them inefficient due to overwhelming calculations at registration center. Additionally, Odelu *et al.* asserted that He et al.'s protocol has drawbacks in login and password change phases and is also prone to impersonation attacks. Jiang et al.'s protocol cannot provide user anonymity and makes it prone to trace attacks. In 2016, Reddy *et al.* [13] cryptanalyzed Lu *et al.*'s protocol and showed the weaknesses such as prone to impersonation attacks, man-in middle attacks, clock synchronization problem, lack of user anonymity and lack of perfect forward secrecy. Then they proposed an improved robust protocol built on elliptic curve cryptography. Most recently, Wang *et al.* proposed another three-factor authenticated key agreement protocol for multi-server environment. However, their protocol also shown the flaws such as lack of anonymity, vulnerability to impersonation attacks, insider attacks and clock synchronization problems.

Table 1. Notations of the Proposed Protocol

U_i	An i^{th} user
AS	Application server
RS	Registration server
ID_U	Identity of U_i
PW_U	Password of U_i
BIO_U	Biometrics of U_i
r_U	A random number of U_i for registration
SID_A	Identity of AS
USK	Secret key chosen by RS for U_i
ASK	Secret key chosen by RS for AS
N_1, N_2	Random numbers generated by U_i and AS
T_R	Number of times of U_i registration
C_U	Total count of the AS assigned to U_i
SK	Session key generated by U_i and AS
$Æ$	An adversary
SC	A Smartcard
$Gen(BIO_U)$	Generation function of biometric keys
$Rep(BIO_U)$	Reproduction function of biometric keys
$h(\cdot)$	A secure one-way hash function
$\oplus$	A bitwise exclusive-OR operation
$\|\|$	The concatenation operation

Our contributions: The keen observation of above related works clearly proves that most of them are vulnerable to impersonation attacks. Therefore, this paper proposes another three-factor authentication protocol for multi-server architecture. Unlike the existing protocols, the proposed protocol attains perfect mutual authentication without sharing its user's credentials throughout any phase. This nature of proposed protocol resists all sorts of attacks and makes it robust compared to other protocols.

Roadmap of the paper—Section 2 presents the proposed scheme. Section 3 portrays security analysis of

the proposed scheme in detail. Section 4 affords performance analysis. At last, Section 5 concludes the paper.

2. The Proposed Protocol

This section proposes three-factor remote mutual authentication with key agreement protocol for multi-server architecture. The proposed protocol comprises three participants: user (Ui), application server (AS), registration server (RS) and seven phases: application server registration phase, user registration phase, login phase, mutual authentication with key agreement phase, password and biometrics changing phase, dynamic server addition phase, and user revocation/re-registration phase. The various notations used in the proposed protocol are listed in Table 1.

2.1. Application Server Registration Phase

In this phase, a new *AS* sends a registration request to the *RS* in order to become an authorized server of the network. The *AS* registration process occurs via a secure channel with following steps:

Step 1: *AS* sends registration request < SID_A > to the *RS*.

Step 2: *RS* computes $K_S = h(\mathrm{SID_A} \,||\, ASK)$ and *RS* stores {SID_A, KS} in its database table T_S.

Step 3: *RS* sends < K_S, $h(ASK)$ > to *AS*, which can be used in further phases of authentication.

2.2. User Registration Phase

A new U_i, who desires to avail the services provided by any *AS* must register with *RS.* Assume that U_i obtains a *SC* with the value {$h(\cdot)$} upon formal request to *RS.* U_i goes after the following steps to register with *RS* via a secure channel as shown in Fig. 1.

Step 1: U_i chooses ID_U, PW_U, and generates a random number $ru \in Z_p^*$. U_i computes $PID_U = h(ID_U \,||\, ru)$, $PWD_U = h(PW_U \,||\, ru)$ and sends a request message < PID_U, PWD_U > to *RS.*

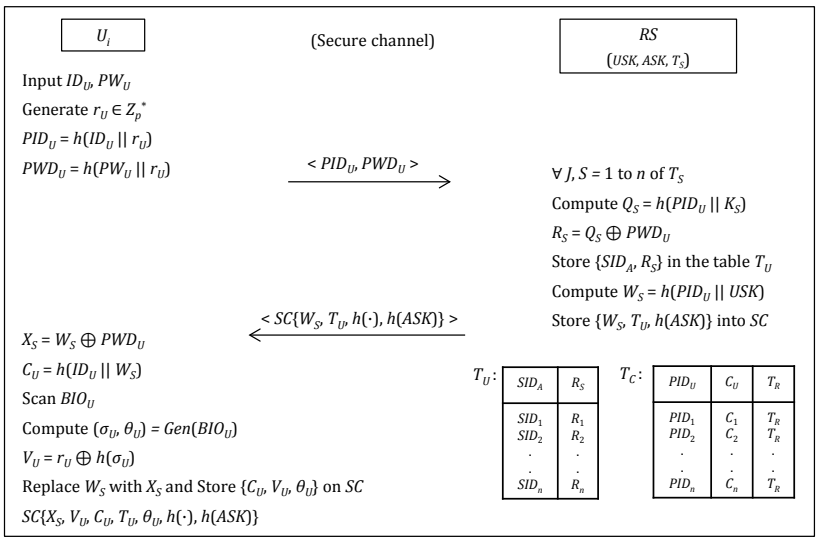

Fig. 1. Summary of user registration phase.

Step 2: *RS* verifies the duplication of PIDU and then computes $Q_S = h(PID_U \,||\, K_S)$, $R_S = Q_S \oplus PWD_U$ for all the registered SID_A and stores {SID_A, R_S} in the table T_U. R_S also stores {PID_U, C_U, $T_R = 1$} in the table T_C, where $T_R = 1$ means U_i is registered once and is in active state. *RS* computes $W_S = h(PID_U \,||\, USK)$ and personalizes {W_S, T_U, $h(ASK)$} into the *SC* to be delivered to U_i.

Step 3: U_i scans his/her BIO_U at the provided sensor with card reading machine and computes $X_S = W_S \oplus PWD_U$, $C_U = h(ID_U \,||\, W_S)$, $(\sigma u, \theta u) = Gen(BIO_U)$, and $V_U = ru \oplus h(\sigma u)$. U_i replaces W_S with X_S and stores {C_U, V_U, θ_U} on the *SC.* Thus the *SC* finally contains the parameters $SC\{X_S, V_U, C_U, T_U, \theta_U, h(\cdot), h(ASK)\}$.

2.3. Login Phase

When a U_i wants to access the services of any registered AS, he/she can launch the login request by inserting SC and inputting ID_U, PW_U and $BIO_{U'}$ as detailed in the Fig. 2.

Step 1: SC computes $\sigma u' = Rep(BIO_{U'}, \theta u)$, $ru = V_U \oplus h(\sigma u')$, $PID_U = h(ID_U \| ru)$, $PWD_U = h(PW_U \| ru)$, $W_S = X_S \oplus PWD_U$ and then verifies whether the condition $C_U \stackrel{?}{=} h(ID_U \| W_S)$ holds. If it generates a negative result, the login request can be terminated. Otherwise, the list of AS appears on the card reading machine.

Step 2: $SC \rightarrow AS$: $M_1 = < B_{US}, D_{US}, N_1 >$

U_i selects the AS he wanted to communicate, then SC retrieves corresponding AS's R_S value from T_U and extracts $Q_S = R_S \oplus PWD_U$. SC generates $N_1 \in Z_p^*$ and computes $Q_S = R_S \oplus PWD_U$, $B_{US} = PID_U \oplus h(SID_A \| N_1 \| h(ASK))$, $D_{US} = h(PID_U \| Q_S \| N_1)$. SC launches the login request message $M_1 = < B_{US}, D_{US}, N_1 >$ to AS.

2.4. Mutual Authentication with Key-Agreement Phase

During this phase, U_i and AS authenticates each other and computes a session key for further secure communication over public channel. The entire mutual authentication with key agreement phase is illustrated in Fig. 2.

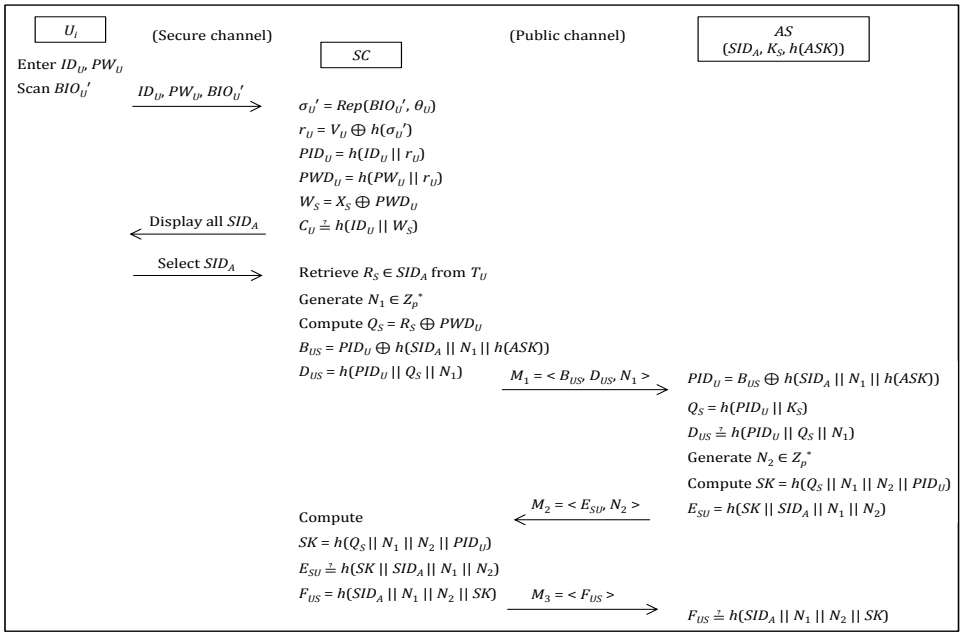

Fig. 2. Summary of login and mutual authentication phases.

Step 1: AS computes $PID_U = B_{US} \oplus h(SID_A \| N_1 \| h(ASK))$, $Q_S = h(PID_U \| K_S)$ and verifies the condition $D_{US} \stackrel{?}{=} h(PID_U \| Q_S \| N_1)$. If the condition holds, then AS can authenticate U_i otherwise the process can be terminated.

Step 2: $AS \rightarrow SC$: $M_2 = < E_{SU}, N_2 >$

AS generates $N_2 \in Z_p^*$ and computes $SK = h(Q_S \| N_1 \| N_2 \| PID_U)$, $E_{SU} = h(SK \| SID_A \| N_1 \| N_2)$, and then sends $M_2 = < E_{SU}, N_2 >$ to SC.

Step 3: $SC \rightarrow AS$: $M_3 = < F_{US} >$

SC computes $SK = h(Q_S \| N_1 \| N_2 \| PID_U)$ and verifies the condition $E_{SU} \stackrel{?}{=} h(SK \| SID_A \| N_1 \| N_2)$. If the condition holds, then U_i can authenticate AS, otherwise the process can be terminated. SC computes $F_{US} = h(SID_A \| N_1 \| N_2 \| SK)$ and sends it to AS.

Step 4: AS verifies $F_{US} \stackrel{?}{=} h(SID_A \| N_1 \| N_2 \| SK)$ and reconfirms the authenticity of U_i. Now, U_i and AS are set to start the communication with the computed session key SK.

2.5. Password and Biometrics Changing Phase

This procedure is invoked when U_i wish to update his/her existing password or biometrics with new ones. In this procedure, U_i can change his password or biometrics by inserting SC and inputting ID_U, PW_U and $BIO_{U'}$ over a secure channel without the help of RS as follows.

Step 1: SC computes $\sigma_U' = Rep(BIO_U', \theta_U)$, $r_U = V_U \oplus h(\sigma_U')$, $PID_U = h(ID_U \| r_U)$, $PWD_U = h(PW_U \| r_U)$, $W_S = X_S \oplus PWD_U$ and then verifies whether the condition $C_U \stackrel{?}{=} h(ID_U \| W_S)$ holds. If it generates a negative result, the login request can be terminated.

Step 2: SC retrieves $R_S = Q_S \oplus PWD_U$ for all SID_A using the current PW_U. U_i chooses a new password $PW_U^{\#}$ and scans new $BIO_U^{\#}$ and then computes $PWD_U^{\#} = h(PW_U^{\#} \| r_U)$, $X_S^{\#} = W_S \oplus PWD_U^{\#}$, $(\sigma_U^{\#}, \theta_U^{\#}) = Gen(BIO_U^{\#})$,$V_U^{\#} = r_U \oplus h(\sigma_U^{\#})$, $R_S^{\#} = Q_S \oplus PWD_U^{\#}$ for all SID_A.

Step 3: U_i updates the table $T_U^{\#}$ and the parameters $X_S^{\#}$, $V_U^{\#}$, $\theta_U^{\#}$ on the SC. Thus, the SC finally consists of the parameters $SC\{X_S^{\#}, V_U^{\#}, C_U, T_U^{\#}, \theta_U^{\#}, h(\cdot), h(ASK)\}$.

2.6. Dynamic Addition of Application Server Phase

In this phase, a new application server AS^{new} can join the existing network by sending a registration request to the RS in order to become an authorized server. The new application server's information will be forwarded to the existing users of the network periodically using their stored $\{PID_U, C_U\}$ or upon the request for updates from U_i. In any of the above two cases, U_i is expected to pass through login phase successfully. The AS^{new} registration process consists of following steps.

Step 1: AS^{new} sends registration request < $SID_A{}^{new}$ > to the RS.

Step 2: RS computes $K_S{}^{new} = h(SID_A{}^{new} \| ASK)$, and RS stores $\{SID_A{}^{new}, K_S{}^{new}\}$ in T_S.

Step 3: RS sends < $K_S{}^{new}$ > to AS^{new} via a secure channel, which can be used in further phases of authentication.

Step 4: RS retrieves the $\{PID_U, C_U\}$ of users and computes $Q_S{}^{new} = h(PID_U \| K_S{}^{new})$, and then delivers it to U_i via a secure channel. RS updates $\{PID_U, C_U\}$ in the table T_C. Upon receiving the new AS's information, U_i computes $R_S{}^{new} = Q_S{}^{new} \oplus PWD_U$ and adds it to their T_U.

2.7. User Revocation/Re-Registration Phase

This phase directs U_i when he/she wants revoke the available services or re-register with different identity. In any of the above two cases, U_i is expected to pass through login phase successfully. Both the phases occurs via a secure channel as explained below.

Revocation: During the revocation phase, U_i proves his/her legitimacy and submits the acquired SC to the RS. Upon receiving the request, RS sets $T_R = 0$ and updates $\{PID_U, C_U, T_R = 0\}$ in the table T_C, where $T_R = 0$ means U_i is revoked and inactive. When U_i wishes to resume his/her previous services, then U_i must prove his/her $PID_U \stackrel{?}{=} h(ID_U \| r_U)$, $PWD_U \stackrel{?}{=} h(PW_U \| r_U)$ by retrieving $Q_S = R_S \oplus PWD_U$ using the last recent ID_U, PW_U, and BIO. If U_i holds the correct credentials, then RS resumes U_i's services and updates $\{PID_U, C_U, T_R = 1\}$ in the table T_C.

Re-registration: When U_i wants to re-register with the new credentials, then he/she must prove his/her legitimacy and send the request to RS. Upon receiving the request, RS follows the steps described in user registration phase and updates $\{PID_U, C_U, T_R = T_R + 1\}$ in the table T_C.

3. Security Analysis

This section demonstrates the security analysis of the proposed protocol by describing each security feature. The main aim of the proposed protocol is to increase the degree of security of communicating messages over public or insecure channels.

Proposition 1. The proposed protocol achieves user anonymity and untraceability.

Proof. The transmitted messages < M_1, M_2, M_3 > between U_i and AS during the login and authentication phases are arbitrary for each session due to their association with the random numbers N_1 and N_2. U_i's original identity ID_U is encapsulated in the form of $B_{US} = PID_U \oplus h(SID_A \,||\, N_1 \,||\, h(ASK))$, where $PID_U = h(ID_U \,||\, r_U)$. The similar approach is followed in case of all the parameters < B_{US}, D_{US}, N_1 >, < E_{SU}, N_2 >, and < F_{US} > and accomplished user anonymity. The proposed protocol also provides another important feature called untraceability. The randomness of all the parameters makes it unidentifiable and untraceable to the adversaries.

Proposition 2. The proposed protocol is secure against replay attacks.

Proof a. $Æ$ may try to establish a new session while impersonating a valid user by replaying the previous transmitted message < M_1 >. However, the proposed protocol can withstand replay attacks using random number N_1 as explained here. During the login and mutual authentication phase, AS receives the message < B_{US}, D_{US}, N_1 > and stores the pair {PID_U, N_1} in its database. If $Æ$ replays the same message < $M_1^{Æ}$ >, AS retrieves {PID_U, $N_1^{Æ}$} and compares with the stored {PID_U, N_1}. When AS finds $N_1^{Æ} == N_1$, then it drops the request and terminate the process. U_i and AS follows the similar way to defy the replay attack on response messages < E_{SU}, N_2 > and < F_{US} >.

Proposition 3. The proposed protocol is secure against stolen smartcard attacks.

Proof. With the hypothesis that $Æ$ can read a SC stored values using various methods as discussed in, this section describes the resistance of the proposed protocol to stolen smartcard attack. Assume that $Æ$ is able to read the stored parameters {X_S, V_U, C_U, T_U, θ_U, $h(\cdot)$, $h(ASK)$} on a stolen legitimate SC. Now, $Æ$ may try either launching an authentication request to gain the access to AS or try deriving actual U_i's credentials from the extracted parameters. However, $Æ$ undeniably cannot perform any of above actions using these values, since all the important parameters such as $R_S = Q_S \oplus PWD_U$, $X_S = W_S \oplus PWD_U$, $V_U = r_U \oplus h(\sigma_U)$ are safeguarded with $h(.)$, where $PID_U = h(ID_U \,||\, r_U)$ and $PWD_U = h(PW_U \,||\, r_U)$. $Æ$ can neither obtain the credentials nor build an authentication request < M_1 > using the stolen SC due to the unavailability of ID_U, PW_U and BIO_U. At the same time guessing the ID_U, PW_U and forging BIO_U are impractical. Therefore, the proposed protocol can withstand smartcard stolen attacks.

Proposition 4. The proposed protocol is secure against user impersonation attacks.

Proof a. Assume a situation where $Æ$ possesses a valid SC and wants to gain network access by perpetrating user impersonation attack. If $Æ$ wants to impersonate a legitimate U_i, he/she requires to build a login request message M_1 = < B_{US}, D_{US}, N_1 >. On the other hand, $Æ$ should undergo login phase before making authentication request. During login phase, SC computes $\sigma_U' = Rep(BIO_U', \theta_U)$, $r_U = V_U \oplus h(\sigma_U')$, $PID_U = h(ID_U \,||\, r_U)$, $PWD_U = h(PW_U \,||\, r_U)$, $W_S = X_S \oplus PWD_U$ and then verifies whether the condition $C_U \stackrel{?}{=} h(ID_U \,||\, W_S)$ holds. Unless the $Æ$ passes the correct credentials, he/she cannot enter into the further phases. Therefore, $Æ$ certainly requires legitimate credentials for any likewise computations. However, the probability of yielding correct ID_U and PW_U is negligible. Though the $Æ$ performs guessing attacks for ID_U and PW_U, he/she definitely cannot forge or copy valid U_i's BIO_U.

Proof b. The proposed protocol does not share much personal identifiable information of U_i to any AS. During login and mutual authentication phase, AS can obtain only PID_U of legitimate U_i via $PID_U = B_{US} \oplus h(SID_A \,||\, N_1 \,||\, h(ASK))$. For instance, if any AS turns as $Æ$ and wants to impersonate a valid U_i, he/she still requires K_S of targeted AS to construct $Q_S = h(PID_U \,||\, K_S)$. In the proposed protocol, Q_S value is unique for each AS, where $K_S = h(SID_A \,||\, ASK)$. Aforementioned constraints prove that our scheme is secure from user impersonation attacks.

Proposition 5. The proposed protocol is secure against application server impersonation attacks.

Proof. Consider a scenario where a registered *AS* turned as *Æ* captures < M_1 > and tries to impersonate valid *AS* by responding with computed message < $M_2^{Æ}$ >. From the captured M_1, *Æ* can barely obtain $PID_U = B_{US} \oplus h(SID_A \| N_1 \| h(ASK))$ and N_1 when it is assumed of having SID_A. In order to compute the response $M_2^{Æ}$, Q_S value of the targeted *AS* and U_i is a prerequisite since each *AS* of the proposed protocol holds unique long-term key K_S which is computed based on SID_A as $K_S = h(SID_A \| ASK)$. Thus, *Æ* cannot compute $SK = h(Q_S \| N_1 \| N_2 \| PID_U)$, $E_{SU} = h(SK \| SID_A \| N_1 \| N_2)$ and reply U_i. Let's take another case where *Æ* computes $SK^{Æ} = h(Q_S^{Æ} \| N_1 \| N_2 \| PID_U)$, $E_{SU}^{Æ} = h(SK^{Æ} \| SID_A \| N_1 \| N_2)$, and then sends $M_2^{Æ} = < E_{SU}^{Æ}, N_2 >$ to *SC*. Upon receiving the response, U_i computes $SK = h(Q_S \| N_1 \| N_2 \| PID_U)$, $E_{SU} = h(SK \| SID_A \| N_1 \| N_2)$ and can identify it as a malicious attempt due to the non-equivalence of messages $E_{SU}^{Æ} \neq E_{SU}$. It is evident from the above statements that the proposed protocol can withstand application server impersonation attacks.

Proposition 6. The proposed protocol is secure against password guessing attacks.

Proof. *Æ* may try to guess the PW_U using the extracted parameters stored on $SC\{X_S, V_U, C_U, T_U, \theta_U, h(\cdot), h(ASK)\}$ or keep trying to login while guessing the PW_U. However, *Æ* cannot validate the guessed PW_U due to non-availability of parameter r_U.

On the other hand, r_U value is protected with U_i's BIO_U in the form of $(\sigma_U, \theta_U) = Gen(BIO_U)$, and $V_U = r_U \oplus h(\sigma_U)$ and it is believed to be impractical to forge a valid U_i's BIO_U. The *Æ* definitely cannot proceed further without passing correct BIO_U resulting in failure of validating the guessed password using $PID_U = h(ID_U \| r_U)$, $PWD_U = h(PW_U \| r_U)$, $W_S = X_S \oplus PWD_U$, $C_U \stackrel{?}{=} h(ID_U \| W_S)$. In this way, the proposed protocol is secure against password guessing attacks.

Table 2. Comparison of Security Properties with Other Three-Factor Protocols

Security Property	Chuang [5] 2014	Mishra [6] 2014	Lu [8] 2015	He [11] 2015	Wang [9] 2016	Proposed Protocol
User anonymity and untraceability	✗	✗	✗	✓	✗	✓
Perfect mutual authentication	✓	✓	✓	✗	✓	✓
Prevent replay attacks	✓	✓	✓	✓	✓	✓
Prevent man-in-middle attacks	✗	✗	✗	✓	✓	✓
Prevent stolen smartcard attacks	✗	✗	✓	✓	✓	✓
Prevent user impersonation attacks	✗	✗	✗	✗	✗	✓
Prevent server impersonation attacks	✗	✗	✗	✓	✗	✓
Prevent insider attacks	✓	✓	✓	✓	✗	✓
Prevent denial-of-service attacks	✓	✓	✓	✓	✓	✓
Prevent password guessing attacks	✓	✓	✓	✓	✓	✓
Prevent clock synchronization problem	✓	✓	✗	✓	✗	✓
Efficient password changing phase	✓	✓	✓	✗	✓	✓
Provides user revocation/re-registration phase	✗	✗	✗	✗	✓	✓

Proposition 7. The proposed protocol is secure against privileged insider attacks.

Proof. During user registration phase of the proposed protocol scenario, U_i does not submit either the plain credentials or the digest of credentials alone to the *RS*. U_i submits $PID_U = h(ID_U \| r_U)$, and $PWD_U = h(PW_U \| r_U)$ to *RS*, where $r_U \in Z_p^*$ is a random number. Thus, an insider cannot obtain the original credentials of any U_i. Additionally, the proposed protocol is designed on the basis of not maintaining password verification tables and the authentication of entities is being done by verifying the accuracy of received messages such as $D_{US} \stackrel{?}{=} h(PID_U \| Q_S \| N_1)$. Therefore, the proposed protocol attains resistance to insider attacks.

Table 3. Comparison of Computation Cost with Other Three-Factor Protocols

Phase	Entity	Chuang [5] 2014	Mishra [6] 2014	Lu [8] 2015	He [11] 2015	Wang [9] 2016	Proposed Protocol
Login & Authentication	U_i	$8T_h$	$10T_h$	$9T_h$	$7T_h + 3T_p$	$8T_h$	$9T_h$
	AS	$8T_h$	$7T_h$	$7T_h$	$6T_h + 3T_p$	$6T_h$	$6T_h$
	RC	—	—	—	$10T_h + 2T_p$	—	—
	Total	$16T_h$	$17T_h$	$16T_h$	$23T_h + 8T_p$	$14T_h$	$15T_h$
Password & Biometrics Changing	U_i	$2T_h$	$5T_h$	$5T_h$	$2T_h$	$4T_h$	$6T_h$
	RC	—	—	—	—	—	—
	Total	$2T_h$	$5T_h$	$5T_h$	$2T_h$	$4T_h$	$6T_h$

4. Performance Analysis

This section demonstrates the comparison between the proposed protocol and other related protocols in terms of various aspects such as security, and computational cost. The performance analysis ensures that the proposed protocol is efficient and better in every aspect when compared to other authentication protocols for multi-server architecture.

Functionality comparison: Here, the proposed protocol is compared with the three-factor authentication protocols such as Chuang *et al*., Mishra *et al*., Lu *et al,* He *et al*., and Wang *et al*., and is showed in Table 2. It is evident from the Table that except the proposed protocol, all the other three-factor protocols are vulnerable to various security attacks whereas the proposed protocol can prevent user and server impersonation attacks, and also provides perfect user anonymity, user revocation and re-registration phase.

Computational cost comparison: To evaluate the computational cost analysis, we give few notations for the involved actions in all the compared protocols such as T_h: Time complexity of a one-way hash function; T_p: Time complexity of a point multiplication operation on elliptic curve; T_f: Time complexity of encryption or decryption function. To evaluate the computational time analysis, we account $T_h \approx 0.0023$ms, $T_p \approx 2.226$ms, $T_f \approx 0.0046$ms as reported in [12].

From the Table 3, it is evident that Chuang *et al*., Mishra *et al*., Lu *et al*., He *et al*., and Wang *et al*., and the proposed protocol's login and authentication phase requires the computation complexity of $16T_h$, $17T_h$, $16T_h$, $23T_h + 8T_p$, $14T_h$, and $15T_h$, respectively. As shown in Table 4, Chuang *et al*., Mishra *et al*., Lu *et al*., He *et al*., and Wang *et al*., and the proposed protocol's login and authentication phase requires the computation time of 0.036ms, 0.039ms, 0.036ms, 17.860ms, 0.032ms, and 0.035ms, respectively. The proposed protocol consumes less computations compared to Chuang *et al*., Mishra *et al*., Lu *et al*., and He *et al*.'s protocols and more computations compared to Wang *et al*.'s protocol.

5. Conclusion

This paper has conducted an extensive study of the existing authentication protocols for multi-server environment that are developed based three-factor methodologies. We have reviewed few of the recently proposed protocols and spotted severe security drawbacks due to the flaws in their designs. Thus, we have proposed another protocol on the key point of achieving mutual authentication with key-agreement without sharing decisive personal identifiable information. The proposed protocol is also a three-factor and light-weight protocol. The comparison and analysis sections of this paper proves that the proposed protocol performs better than other related protocol.

Acknowledgment

This work was supported by the BK21 Plus project (SW Human Resource Development Program for Supporting Smart Life) funded by the Ministry of Education, School of Computer Science and Engineering, Kyungpook National University, Korea (21A20131600005).

References

[1] Yang, D., & Yang, B. (2010). A biometric password-based multi-server authentication scheme with smart card. *Proceedings of International Conference on Computer Design and Applications: Vol. 5.* (pp. 5-554).

[2] Yoon, E. J., & Yoo, K. Y. (2013). Robust biometrics-based multi-server authentication with key agreement scheme for smart cards on elliptic curve cryptosystem. *The Journal of Supercomputing, 63(1),* 235-255.

[3] He, D. (2011). Security flaws in a biometrics-based multi-server authentication with key agreement scheme. *IACR Cryptology Print Archive,* 365.

[4] Kim, H., Jeon, W., Lee, K., Lee, Y., & Won, D. (2012). Cryptanalysis and improvement of a biometrics-based multi-server authentication with key agreement scheme. *Proceedings of International Conference on Computational Science and Its Applications* (pp. 391-406). Berlin Heidelberg: Springer.

[5] Chuang, M. C., & Chen, M. C. (2014). An anonymous multi-server authenticated key agreement scheme based on trust computing using smart cards and biometrics. *Expert Systems with Applications, 41(4),* 1411-1418.

[6] Mishra, D., Das, A. K., & Mukhopadhyay, S. (2014). A secure user anonymity-preserving biometric-based multi-server authenticated key agreement scheme using smart cards. *Expert Systems with Applications, 41(18),* 8129-8143.

[7] Lin, H., Wen, F., & Du, C. (2015). An improved anonymous multi-server authenticated key agreement scheme using smart cards and biometrics. *Wireless Personal Communications, 84(4),* 2351-2362.

[8] Lu, Y., Li, L., Yang, X., & Yang, Y. (2015). Robust biometrics based authentication and key agreement scheme for multi-server environments using smart cards. *PloS One, 10(5).*

[9] Wang, C., Zhang, X., & Zheng, Z. (2016). Cryptanalysis and improvement of a biometric-based multi-server authentication and key agreement scheme. *Plos One, 11(2).*

[10] Jiang, P., Wen, Q., Li, W., Jin, Z., & Zhang, H. (2015). An anonymous and efficient remote biometrics user authentication scheme in a multiserver environment. *Frontiers of Computer Science, 9(1),* 142-156.

[11] He, D., & Wang, D. (2015). Robust biometrics-based authentication scheme for multiserver environment. *IEEE Systems Journal, 9(3),* 816-823.

[12] Odelu, V., Das, A. K., & Goswami, A. (2015). A secure biometrics-based multi-server authentication protocol using smart cards. *IEEE Transactions on Information Forensics and Security, 10(9),* 1953-1966.

[13] Reddy, A. G., Das, A. K., Odelu, V., & Yoo, K. Y. (2016). An enhanced biometric based authentication with key-agreement protocol for multi-server architecture based on elliptic curve cryptography. *PloS One, 11(5).*

[14] Hsiang, H. C., & Shih, W. K. (2009). Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment. *Computer Standards & Interfaces, 31(6),* 1118-1123.

[15] Juang, W. S. (2004). Efficient multi-server password authenticated key agreement using smart cards. *IEEE Transactions on Consumer Electronics, 50(1),* 251-255.

[16] Leu, J. S., & Hsieh, W. B. (2014). Efficient and secure dynamic ID-based remote user authentication scheme for distributed systems using smart cards. *IET Information Security, 8(2),* 104-113.

[17] Li, X., Ma, J., Wang, W., Xiong, Y., & Zhang, J. (2013). A novel smart card and dynamic ID based remote user authentication scheme for multi-server environments. *Mathematical and Computer Modeling, 58(1),*

85-95.
[18] Liao, Y. P., & Wang, S. (2009). A secure dynamic ID based remote user authentication scheme for multi-server environment. *Computer Standards & Interfaces, 31(1),* 24-29.
[19] Reddy, A. G., Das, A. K., Yoon, E. J., & Yoo, K. Y. (2016). An anonymous authentication with key-agreement protocol for multi-server architecture based on biometrics and smartcards. *Ksii Transactions on Internet and Information Systems, 10(7),* 3371-3396.
[20] Sood, S. K., Sarje, A. K., & Singh, K. (2011). A secure dynamic identity based authentication protocol for multi-server architecture. *Journal of Network and Computer Applications, 34(2),* 609-618.
[21] Tsai, J. L. (2008). Efficient multi-server authentication scheme based on one-way hash function without verification table. *Computers & Security, 27(3),* 115-121.
[22] Xue, K., Hong, P., & Ma, C. (2014). A lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture. *Journal of Computer and System Sciences, 80(1),* 195-206.

Alavalapati Goutham Reddy received his master degree in computer science and engineering from Christ University, India in the year 2013. He is currently a Ph.D. student at School of Computer Science and Engineering at Kyungpook National University, South Korea. His primary research interests revolve around cryptographic authentication protocols and information security. He is a student member of IEEE and ACM.

Eun-Jun Yoon received his Ph.D. degree in computer engineering from Kyungpook National University, South Korea in the year 2006. He is now a professor at Department of Cyber Security, Kyungil University, South Korea. His research interests are cryptography, authentication technologies, smart card security, multimedia security, network security, mobile communications security, and steganography. He has published 75 conference proceedings and 50 journal publications.

Young-Ju Kim received his B.S. degree in computer science and engineering from in the year 2015. He is currently pursuing his M.S. degree at School of Computer Science and Engineering at Kyungpook National University, South Korea. His primary research interests revolve around steganography.

Kee-Young Yoo received his M.S. degree in computer engineering from KAIST, Korea in 1978 and Ph.D. degree in Computer Science from Rensselaer Polytechnic Institute (RPI), U.S.A in 1992. Currently, he is a professor at School of Computer Science and Engineering at Kyungpook National University, South Korea. His area of expertise includes cryptography, steganography, wireless mesh network and RFID security. He is author of more than 200 conference proceedings and 195 journal publications.